AFRL-RI-RS-TR-2018-299

SELF-TIMED ROUTER FOR A NATIVE TRUENORTH LINK

CORNELL UNIVERSITY

DECEMBER 2018

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2018-299 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
THOMAS RENZ for
UTTAM MAJUMDER
Work Unit Manager

/ S /
JOHN MATYJAS
Technical Advisor, Computing
& Communications Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
DECEMBER 2018	FINAL TECHNICAL REPORT	AUG 2015 – JUN 2018

4. TITLE AND SUBTITLE

SELF-TIMED ROUTER FOR A NATIVE TRUENORTH LINK

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA8750-15-1-0173

5c. PROGRAM ELEMENT NUMBER
62788F

6. AUTHOR(S)

Rajit Manohar

5d. PROJECT NUMBER
95SB

5e. TASK NUMBER
CR

5f. WORK UNIT NUMBER
NL

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Cornell University
Office of Sponsored Programs
373 Pine Tree Road
Ithaca, NY 14850-2820

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RITB
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2018-299

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The TrueNorth neuromorphic chip has demonstrated significant advantages in power efficiency compared to traditional processors and graphics processing units at tasks related to image and audio classification. However, while the chip has extremely low power consumption, its non-standard I/O interface requires conversion circuitry that currently dominates the overall system power consumption. This report summarizes the work conducted to eliminate this overhead, and provide a native interface to the TrueNorth chip with low power consumption.

15. SUBJECT TERMS
Self Timed Circuits, Spike Ports, Spike-In/Spike-Out Communication TrueNorth Chip

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			**UTTAM MAJUMDER**
U	U	U	UU	19	19b. TELEPHONE NUMBER (*Include area code*) **NA**

Table of Contents

List of Figures

1. SUMMARY

A six-year collaborative research effort between IBM Research and Prof. Manohar's research group led to the development of the TrueNorth architecture, a programmable hardware platform for neuromorphic computing. TrueNorth is a single-chip million neuron, 256 million synapse chip with a power budget of 70 mW—significantly lower than other neuromorphic systems [1]. TrueNorth takes the neuromorphic paradigm to its limit—all primary inputs and outputs are encoded as spike trains and communicated via a self-timed routing interface. Using conventional clocked logic to implement this interface can lead to overheads, because sequencing in clocked logic is typically performed at the granularity of individual clock cycles—many more gates than the sequencing that can be implemented with self-timed logic. The research effort described in this report developed a native hardware interface to the TrueNorth chip that simplify future integration efforts.

2. INTRODUCTION

2.1 The TrueNorth chip

The TrueNorth chip consists of an array of neurosynaptic cores. Each neurosynaptic core consists of 256 neurons and 256 axons. Every input axon can be connected to any subset of the 256 neurons in the core. A spike produced by neuron in a core can target any axon in any neurosynaptic core, within a certain range (determined by the number of bits in the packet routing information). A single TrueNorth chip contains 64x64 array of these neurosynaptic cores.

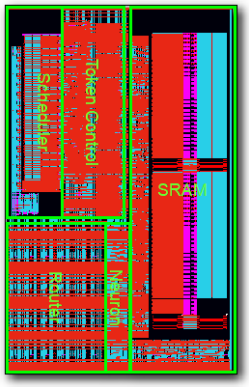

Figure 1. A single neurosynaptic core from TrueNorth, showing the self-timed on-chip router. The chip consists of a 64x64 array of identical cores.

TrueNorth's on-chip network uses a bidirectional mesh routing architecture, with a dimension-ordered deterministic packet router implemented with self-timed logic.

Figure 1 shows a tile from the TrueNorth layout, with the router on the bottom left corner. Packets contain routing information represented as a pair (dx, dy), where "dx" and "dy" indicate the distance the packet must travel in the x- or y-direction to reach its destination. The packet also contains two additional pieces of information: (i) an 8-bit destination axon address, specifying which axon in the target core receives the spike, and (ii) a 4-bit value specifying the axonal delay, which determines the global 1 KHz tick at which the spike must be delivered to the destination axon.

The edge of the chip, therefore, contains 64 bidirectional communication links connected to the mesh routing fabric. To minimize the number of I/O pads, a collection of these links are combined using a non-deterministic merge (a "funnel") into a single chip-to-chip link that contains both the packet information as well as an identifier specifying the index of the on-chip communication channel on which the packet was received. This identifier is used by the receiver to de-multiplex the chip-to-chip communication link (using a "horn"), sending the packet to the appropriate on-chip communication link on the next TrueNorth chip. The goal of this effort is to create an interface to this router for Air Force Rome Laboratory (AFRL), so that AFRL chips can be a part of the TrueNorth fabric.

2.2 Interface Description

The external communication interface to the TrueNorth chip consists of a self-timed data link that connects to the array of neurosynaptic cores through a funnel and horn circuit (Figure 2). The self-timed link uses a bundled-data protocol for communication, along with a two-phase signaling protocol that alternates even and odd bits on the same data bundle. The TrueNorth chip is asynchronous, and hence the chip can accept an input handshake at any time as well as produce an output handshake at any time, so long as the

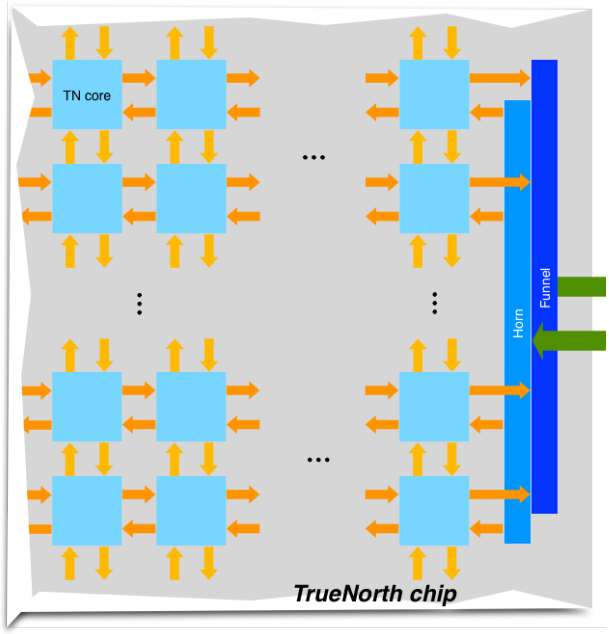

Figure 2. Block diagram of the internals of the TrueNorth chip, showing the external interface including the funnel and horn.

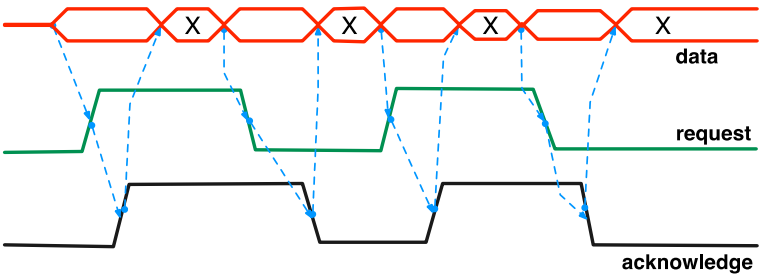

Figure 3. Signaling convention for a two-phase bundled data interface.

environment connected to the chip respects the control handshaking protocol. Figure 3 summarizes the signaling convention used for two-phase bundled data communication protocols. The blue arrows depict ordering constraints between different signal transitions.

Most off-the-shelf electronics operate using the synchronous paradigm. This means that all state changes occur at deterministic points in time governed by the timing specified by a clock signal. In the rest of this report, we assume this is the desired environment interface because almost every commercial and research chip uses this approach today.

A single data transfer using the two-phase handshake protocol used by TrueNorth requires two sequential steps. A naïve synchronous implementation would require two clock cycles for this operation, which halves the communication throughput. This effect can be mitigated by using a half-cycle implementation that uses both clock edges for sequencing. Such an interface could communicate effectively with TrueNorth, except for the fact that the acknowledge signal from the TrueNorth chip could change at any time relative to the clock edge used in the interface electronics. Because of the unconstrained nature of this signal, it is possible that the interface electronics might have setup/hold time violations, causing the synchronous interface electronics to malfunction.

To mitigate this issue, a standard approach is to use synchronizers [2]. Doing so would add two flip-flops to the acknowledge signal path to reduce the probability of interface failure [3]. Unfortunately this results in a significant reduction in throughput of the interface, since changes in the acknowledge signal will now require multiple clock cycles to propagate to the state machine in the interface electronics.

3. METHODS, ASSUMPTIONS, AND PROCEDURES

3.1 Interface Assumptions and Decomposition

In consultation with Air Force Research Laboratory (AFRL), we developed a number of goals for a standardized interface to the TrueNorth chip. The major goal was to eliminate the use of additional "glue" logic to connect to TrueNorth for the most common use cases. The two major use cases that were determined include:

- Communication with a standard desktop/laptop host computer. In this mode, data would be transferred from the host computer to/from the TrueNorth chip over the Universal Serial Bus (USB) interface. The interface electronic would convert to and from the USB format to the native TrueNorth format.
- Communication with a standard component like a video camera that has a USB output. In this mode, the interface would transfer data from the external USB device and send it to the TrueNorth chip directly.

These two modalities require different types of USB operation. Since the USB standard is evolving with the current generation supporting higher data rates compared to when this project started, we opted to partition the problem of USB interfacing into two components: (a) a commercial, off-the-shelf part that could be upgraded/modified as the USB interface evolved; and (b) a custom designed circuit that supports interfaces to a standard parallel, low frequency first-in first-out (FIFO) interface that could easily be generated using simple firmware on commercially available USB interface electronics.

Hence, the overall interface design was partitioned into three pieces:

- A firmware component, where a commercial, off-the-shelf USB interface part was programmed/configured to convert the USB interface into a standard parallel synchronous data bus that has a FIFO protocol. The component we used to demonstrate this functionality was the Cypress FX3 USB interface chip.
- A synchronous hardware component that can access the parallel synchronous FIFO and perform any data format/framing required for communicating with the TrueNorth chip.

- A synchronizer circuit that converts the parallel synchronous data interface into a self-timed protocol that corresponds to the native TrueNorth interface.

3.2 Firmware Design

The Cypress FX3 chipset has a USB 3.x interface combined with a number of standard peripheral interfaces such as a serial peripheral interface (SPI) and a Universal Asynchronous Receive/Transmit (UART) interface. To control the interfaces and orchestrate data movement, it has an integrated ARM microprocessor. Cypress provides a number of reference firmware designs that demonstrate the capabilities of its chipset, and these were leveraged to create the firmware for this project.

Two firmware mechanisms were developed: one that could communicate with a USB device like a camera, and a second that could communicate with a host computer. Both of these used existing templates provided by Cypress. The functionality in the firmware simply translates the data received to/from the USB interface from/to the wire parallel output interface provided by the FX3 chipset.

To communicate with the Cypress FX3, Python code was developed to send and receive data over USB from a host computer. This task included developing a packet format that could be decoded by the framing module (described in Section 3.3) to appropriately deliver spikes to the TrueNorth chip.

3.3 Framing Data

One of the issues with interfacing to the TrueNorth chip is in data representation. As a concrete example, a single color channel of an image might be represented using eight-bit precision as an integer in the range [0,255]. However, the native data format in TrueNorth is in spikes—essentially a unary representation. To reduce the power consumption, it is best that the conversion between the compact binary representation and the much larger unary representation be performed as late as possible prior to interfacing with the TrueNorth chip. Hence, this conversion is performed after the USB data has been read and during the operation of the synchronizer.

To set up the conversion procedure, configuration commands can be transmitted over the USB interface. These commands can be used to set up conversion in the following formats:

- k-out of-N unary spike train: In this approach, a value of N can be pre-configured, and data is assumed to be in the range 0 to N. When a data value k arrives, it is converted to k spikes that are provided on successive time slots. The next data value will only be transmitted after N time slots have elapsed. In this approach, all data packets transmitted are expanded into N TrueNorth time slots. This corresponds to using deterministic rate encoded spikes, a common way to represent data in TrueNorth.
- at-k out-of-N spike: In this approach, the value of N is pre-configured as usual. However, the way data is represented is different. When a data value k arrives, a single spike is generated at time slot k. A value of zero is indicated by no spikes. Once again, data packets are expanded into N TrueNorth time slots. This is sometimes referred to as a time-to-spike code, and uses much less power compared to rate encoding; however, it is more challenging to compute using this code versus rate encoding.

Finally, spikes received from the TrueNorth chip can also be re-encoded into standard binary formats using the same conversion (except in the opposite direction).

These conversion methods were implemented using the Verilog hardware description language, which can be converted into a synchronous hardware implementation using standard commercial tools.

3.4 Native Interface

There are several options when designing a synchronous interface that can communicate with a spike-based self-timed protocol. The interface has to have two properties: provide a simple communication protocol with external synchronous logic, and provide a high-throughput interface to the self-timed TrueNorth communication link. Since the self-timed link has a request-acknowledge protocol, the synchronous link must also have a flow-control mechanism. This requirement is compatible with the way the standard USB protocol operates.

The second major piece of the interface is the conversion of a clocked interface to one that is self-timed. The classic problem of re-synchronizing an asynchronous signal to a clock is known to exhibit the phenomenon of *metastability*. There are well-known solutions to this problem that have the property that the probability that the circuit malfunctions decays exponentially with the amount of time allocated for the decision-making process. Hence, simply waiting long enough is sufficient.

The most common synchronizer is the "two flip-flop" synchronizer shown in Figure 4. In this design, the transmitted signal to be synchronized is passed through two sequential flip-flops both of which are clocked by the receiver's clock [3]. The presence of two flip-flops ensures that at least a full cycle is permitted for the resolution of any metastability. Unfortunately while this solution addresses the metastability issue, converting this into a full interface results in low throughput operation as discussed earlier.

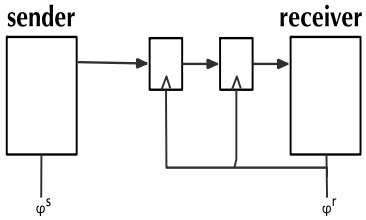

Figure 4. A classic, two flip-flop synchronizer for communication between two different synchronization domains.

Instead, we used a significantly higher throughput interface based on the principle of *gradual* synchronization [4]. In this approach, a multi-stage synchronization FIFO is constructed, with the probability of metastability dropping exponentially with the number of stages. Computation can be hidden behind synchronization, enabling any logic required for the interface to TrueNorth to be hidden behind the synchronization latency. The high-level view of this interface is illustrated in Figure 5.

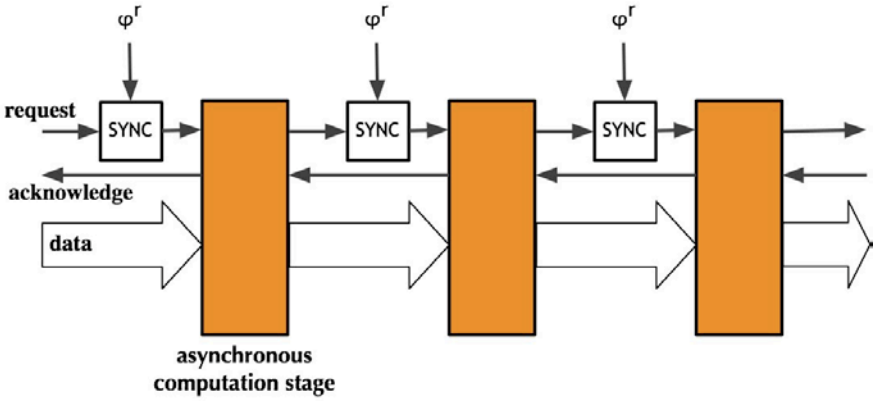

Figure 5. Gradual synchronization: pipeline synchronization combined with computation.

To demonstrate this interface, we developed a custom VLSI circuit implementation of the synchronization hardware. The synchronization hardware takes an interface clock signal as an input, and produces an output self-timed interface that is compatible with the TrueNorth spiking interface. The interface was designed using the asynchronous circuit toolkit (ACT), which enables the design of parameterized asynchronous circuits. The interface is parameterized in two ways: (a) the number of pipeline stages used for synchronization, and (b) the bit-width of the data being communicated through the gradual synchronization pipeline. The depth of the pipeline determines the latency of the interface; as the latency goes up linearly with the pipeline depth, the probability of synchronization failure drops exponentially.

The ACT circuits were automatically converted into a SPICE format circuit that can be processed using commercial tools. For example, the ACT circuits were extensively simulated with commercial tools to verify their functionality. We had the opportunity to incorporate the ACT circuits in a silicon implementation for a different project, and so we completed the physical design of the interface electronics and included it in a fabricated chip for additional verification of its functionality.

4. RESULTS AND DISCUSSION

4.1 End-to-end Evaluation

To complete an end-to-end demonstration of the different components of the interface, the Python interface was run on a Linux-based laptop. A prototyping board for the Cypress FX3 chipset was used to connect to the USB interface on the laptop, and the parallel interface from the Cypress evaluation board was connected by wires to a field-programmable gate array (FPGA) prototyping board. The FX3 board was initialized with the firmware for communication with the host computer, and the FPGA was used to prototype the Verilog implementation of the packet to/from spike interface.

Since we did not have a breakout board with the TrueNorth chip, we used Verilog to implement simple spiking neuron functionality and used it for an end-to-end demonstration.

We tested stimulating neurons via spike trains, receiving values from the neuron and converting them into values with a rate-encoded conversion mechanism. All the tests performed were successful. For completeness, we also included a Video Graphics Array (VGA) output directly from the FPGA to visualize the spike delivery to an array of neurons so that we could visually verify the functionality of the interface design. Tested scenarios included small multi-layer neural network with adjacent layers being fully connected, as well as adjacent layers having a connectivity structure resembling a convolutional neural network.

One of the issues raised by the evaluation is the generality of the data conversion interface. The current interface has two different spike to/from data formats, along with a raw interface that simply assumes that the input from the USB interface is in fact a spike. The data converter interfaces will naturally have limits on the number of bits in the representation, the way spikes are framed for the TrueNorth chip, etc. However, these parameters are not set in stone in general. The TrueNorth chip is extremely flexible, and users could pick completely different data encodings or time slot mechanisms. There is a trade-off between the generality of the interface designed and the potential for an interface that does not meet the needs of users. While the current design includes

the representations that are known to be used, a more exhaustive catalog of data representations is needed prior to converting the design into an application-specific integrated circuit (ASIC).

A second issue raised by the architecture is the reliance on a commercial USB interface chip. While not ideal (since these chips use tens or sometimes hundreds of mWatts of power, which is comparable to the 70mW budget of TrueNorth), the requirement that the USB interface handle both a host computer interface as well as an external camera means that significant flexibility is needed in the USB implementation. This flexibility is handled in commercial chips by a microprocessor, and this comes at a high power cost relative to the TrueNorth chip. Removing the commercial USB chip requires a significantly more complex interface ASIC, because it would have to integrate the USB physical layer and protocol stack, as well as include some flexibility (likely a microcontroller) to support the different USB interface modalities needed. This is a much larger project than was envisioned by this effort, but doing this would further reduce the power requirements for interfacing with TrueNorth.

4.2 Synchronizer Evaluation

We performed extensive simulations of the synchronizer circuit, including careful sweeps of asynchronous signals so that they changed very close to clock edges. We verified that the synchronizer circuit would enter a metastable state, but then eventually exit this state as predicted by theory and analysis [4].

A prototype synchronizer was included in a test chip for a different research project, and was found to be functional and successful at mitigating the problem of synchronization failure. However, during experimental testing, it was discovered that the clock lines used by the synchronizer did not have sufficient amplification, and this led to scenarios where data corruption could occur between adjacent data packets. The design of the synchronizer was modified to correct this issue, and we have high confidence in the new design as a result of this experimental evaluation.

We completely separated the synchronizer from the logic required for spike generation. A future design could achieve better performance if the two were integrated. We have an initial design for such an implementation, but more work would be necessary to convert it into a circuit that could be included in a future ASIC.

5. CONCLUSIONS

We developed the components necessary to build an efficient interface to the TrueNorth chip. It consists of three components: (a) firmware for a commercial USB interface chip, and the associated Python program to communicate with the firmware over USB; (b) a Verilog implementation of a data/packet converter from normal binary representation to/from spike-based representations used by TrueNorth; and (c) a high-throughput synchronizer that converts the standard FIFO synchronous interface into an asynchronous handshake protocol suitable for interfacing with the TrueNorth pins. To convert this design into a physical chip, two items are necessary. First, a detailed pin description of the TrueNorth chip is required so that an appropriate interface board be designed. Second, a better understanding of the details of the TrueNorth configuration interface is needed so that an interface board can be designed to include both the spike and configuration interface. Both of these can be achieved via continued collaboration with IBM Research.

6. REFERENCES

1. Paul A. Merolla, John V. Arthur, Rodrigo Alvarez-Icaza, Andrew S. Cassidy, Jun Sawada, Filipp Akopyan, Bryan L. Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, Bernad Brezzo, Ivan Vo, Steven K. Esser, Rathinakumar Appuswamy, Brian Taba, Arnon Amir, Myron D. Flickner, William P. Risk, Rajit Manohar, and Dharmendra Modha. **A Million Spiking-Neuron Integrated Circuit with a Scalable Communication Network and Interface.** *Science*, **345**(6197):668--673, August 2014.
2. T.J. Chaney and C.E. Molnar. **Anomalous Behavior of Synchronizer and Arbiter Circuits**. *IEEE Trans. on Computers* **C-22**(4):421—422, Apr. 1973.
3. Cadence Design Systems. **Clock Domain Crossing: Closing the Loop on Clock Domain Functional Implementation Problems.** *Technical white paper*, 2004.
4. Sandra J. Jackson. **Gradual Synchronization.** Ph.D. thesis, Cornell University, 2014.

7. LIST OF ACRONYMS

Acronym	Expanded Form
ACT	Asynchronous circuit toolkit
AFRL	Air Force Research Laboratory
ASIC	Application-specific integrated circuit
FIFO	First-in first-out
FPGA	Field-programmable gate array
UART	Universal Asynchronous Receive/Transmit
USB	Universal Serial Bus
VGA	Video graphics array